

## REMARKS/ARGUMENTS

Claims 1-30 are in the case. The applicants have studied the office action dated July 11, 2008 and believe the application is in condition for allowance. Reconsideration and reexamination are respectfully requested.

Claims 1-3, 11-13 and 21-23 have been rejected under 35 U.S.C. 103(a) as being unpatentable over Porterfield (Patent No.: US 6480951 B2), hereinafter "Porterfield," and Tang et al (Patent No.: US 6298371 B1), hereinafter "Tang," further in view of Siddabathuni (Patent No.: US 7290038 B2), hereinafter "Siddabathuni". Claims 4-7, 10, 14-17, 20, 24-27 and 30 have been rejected under 35 U.S.C. 103(a) as being unpatentable over Porterfield, Tang and Siddabathuni as applied to claims 1-3, 11-13 and 21-23 above further in view of Applicant admitted prior art, hereinafter, "AAPA." Claims 8-9, 18-19 and 28-29 have been rejected under 35 U.S.C. 103(a) as being unpatentable over Porterfield, Tang and Siddabathuni further in view of Dunham (Patent No.: US 6269431 B1), hereinafter "Dunham". These rejections are respectfully traversed.

Claim 1, for example, is directed to a "method for sending data from a source to a destination, comprising: a host of the source providing to a sending agent of the source, virtual memory addresses of data to be sent to a destination wherein the data is stored in a plurality of unpinned physical locations of the source, each location having a physical address and a virtual memory address which is mapped to the physical address; the sending agent providing to the host of the source at least some of the virtual memory addresses of the data to be sent to the destination; the host of the source identifying to the sending agent the data addressed by the virtual memory addresses provided by the sending agent; and the sending agent sending the identified data to the destination." As set forth in greater detail in the specification, such an arrangement can significantly reduce the amount of source physical memory which is pinned for data transmission by the source as compared to that of the AAPA. It is appreciated that other features may be obtained, depending upon the particular application.

In the November 29, 2007 office action, the Examiner concedes that the Porterfield reference is "silent on disclosing explicitly ... 'the sending agent providing to the host at least some of the virtual memory [addresses] of the data to be sent to the destination.'" However, in the July 11, 2008 office action, the Examiner "points to Porterfield, Col. 3, lines 35-65 and more

specifically in lines 54-60, where translator 205 makes a request to memory through switch 201, seeking a physical address that corresponds to a virtual address received from original source device ...”

The applicants respectfully disagree. It is respectfully submitted that the Examiner has cited no portion of the Porterfield reference teaching or suggesting that data addressed by the virtual addresses is “stored in a plurality of unpinned physical locations of the source” as required by claim 1. On the contrary, it is believed that the cited data is stored in *pinned* locations in the Porterfield reference because the Address Translator 205 translates the addresses using a remapping table *stored* in main memory 125. Porterfield, col. 3, lines 54 et seq. It is submitted that swapping unpinned memory locations could invalidate the stored remapping table.

It is further the Examiner’s position in the November 29, 2007 office action that “Tang discloses ... the sending agent providing to the host at least some of the virtual memory [addresses] of the data to be sent to the destination” citing “(Tang, Col.73, lines 12-22, where host is exposed to all virtual address space, which means agent has provided full range of virtual addresses to the host).” The applicants respectfully disagree.

The Examiner has cited a “VSP wrapper” of the Tang reference which apparently is “a logic wrapper around a digital signal processor (DSP) core, that interfaces the DSP with the PC via the PCI/AGP Bus or PC system core logic.” Tang, col. 15, lines 30 et seq. In performing a transfer, it is respectfully submitted the VSP wrapper as cited does not provide “to the host of the source at least some of the virtual memory addresses of the data to be sent to the destination” as required by claim 1. Furthermore, it is respectfully submitted that the host of the source does not “[identify] to the sending agent the data addressed by the virtual memory addresses provided by the sending agent” as required by claim 1. Instead, the Tang reference makes clear that the VSP wrapper performs data transfers *without host intervention*:

The wrapper acts as a scatter-gather bus master and I/O accelerator by itself that boosts throughput of a multitasking system (even without a DSP chip or core) by relieving the host of I/O chores and providing byte channeling of 32-bit Dword host data into byte-aligned 16-bit VSP word format without host or VSP intervention. The wrapper also has a memory buffer for modem, voice/telephony and audio data. With a DSP, the VSP wrapper can “walk” the entire virtual memory space of the host memory system *without host intervention* thereby making the VSP a super bus master with virtual memory

addressing capability beyond simple scatter-gather bus mastering. With a DSP, the VSP wrapper can further create ping-pong and circular buffers to advantageously unify the buffers currently used in modem, voice and audio applications by replacing modem, voice/telephony and audio add-in cards with the VSP circuitry. Tang, col. 22, lines 1 et seq. [emphasis added]

The VSP apparently converts virtual addresses to physical addresses by using Windows resources:

USP utilizes this fundamental memory management scheme to make a VSP an extension of the host CPU and to share host system memory and resources. USP provides a method for the VSP to grab memory object handles. *Since Windows provides OS services for ascertaining the physical addresses of memory objects when they are locked down, the VSP grabs these handles by Direct DSP software operations that obtain the physical addresses of these handles through Windows and pass them on to the VSP.* With these physical addresses, the VSP accesses memory objects (e.g. via the PCI bus) with VSP acting as a super busmaster for scatter-gather DMA transactions within the entire host accessible virtual memory space. The host CPU/MMX has elaborate paging hardware on-chip for accessing 64 T bytes of virtual memory. VSP conveniently traverses the host virtual memory space as a super busmaster by using these handles (translated to physical addresses) provided by host and OS enhanced with DirectDSP operations. Tang, col. 24, lines 36 et seq. [emphasis added].

In the arrangement of the Examiner's citations to the Tang reference, it is believed that substantial portions of the memory are "locked down" to permit the VSP wrapper to traverse that locked down memory without host intervention. By comparison, as set forth in the present application, an embodiment in accordance with claim 1 can reduce the amount of pinned memory:

[11] In accordance with one embodiment which can improve management of memory resources during data transmission, the host 130 sends to the sending agent 132 the virtual memory addresses of data to be sent. As previously mentioned, the host 130

may include the operating system or a driver, or both. The sending agent 132 may be implemented with a TOE, a network interface card or integrated circuit, a driver, TCP/IP stack, a host processor or a combination of these elements. When the sending agent 132 is ready to send data in either a transmission or alternatively a retransmission, the sending agent 132 provides to the host the virtual addresses of the data it is ready to send, which can be just a portion of the entire datastream which is to be sent. In response, the host provides to the sending agent 132 the physical addresses of the requested data or the actual data itself. As a result, pinning of physical memory can be reduced or eliminated as explained in greater detail below. Present application, paragraph [0011].

In the July 11, 2008 office action, the Examiner “points to Tang, Fig.1, element-110, where unpinned can merely be related to as unlocked physical location, beside it is addressed in AAPA section of instant application that data is stored in a plurality of physical location of the memory (Fig.2, element-52, 10a, 10b, 10c).” The applicants respectfully disagree.

The Tang reference makes clear that the memory accessed by the VSP is locked, not unlocked:

When a memory object is allocated, a handle, rather than a pointer, is generated to identify and to refer to the memory object. The handle is used to retrieve the current address of the allocated memory object. For example, a source handle references a source memory buffer. Processing puts data in a destination memory buffer which is referenced by a destination handle. *When a task needs to access the memory object, the handle for that memory object is preferably locked down. The action of locking down a memory handle temporarily fixes the address of the memory object and provides a pointer to its beginning. While a memory handle is locked, Windows cannot move or discard the memory object. After the object is accessed or the object is not in use, the object handle is then unlocked to facilitate Windows memory management.*

USP utilizes this fundamental memory management scheme to make a VSP an extension of the host CPU and to share host system memory and resources. USP provides a method for the VSP to grab memory object handles. Since Windows provides OS services for ascertaining the physical addresses of memory objects when they are *locked*

down, the VSP grabs these handles by Direct DSP software operations that obtain the physical addresses of these handles through Windows and pass them on to the VSP. With these physical addresses, the VSP accesses memory objects (e.g. via the PCI bus) with VSP acting as a super busmaster for scatter-gather DMA transactions within the entire host accessible virtual memory space. The host CPU/MMX has elaborate paging hardware on-chip for accessing 64 T bytes of virtual memory. VSP conveniently traverses the host virtual memory space as a super busmaster by using these handles (translated to physical addresses) provided by host and OS enhanced with DirectDSP operations. Tang, col. 24, lines 21 et seq. [emphasis added].

Similarly, contrary to the Examiner's assertion the AAPA teaches:

In addition, the host through the host operating system “pins” (block 72) the physical memory locations containing the datastream 10.

For the above reasons, it is clear that the Porterfield, and Tang references do not teach or suggest “the host of the source identifying to the sending agent the data addressed by the virtual memory addresses provided by the sending agent” as required by claim 1. It is further the Examiner's position that Siddabathuni discloses “the host identifying to the sending agent the data addressed by the virtual memory addresses provided by the sending agent” citing “Siddabathuni, Fig. 2 and Fig. 3, Col. 3, lines 58-67, where host identifies the virtual address by means of removing all of its local storage that was mapped to the virtual address space before HCA can tear down a virtual address space, where HCA can be an agent).” The applicants respectfully disagree.

It is respectfully submitted that the Siddabathuni reference makes clear that the cited HCA itself “maps the host storage to InfiniBand virtual address space.” Siddabathuni, col. 3, lines 22 et seq. Furthermore, the virtual addresses are for use by a target device external to the source:

[A] target device may directly access (i.e., via RDMA) host or kernel buffers by writing to or reading from a virtual address mapped to the buffers. Siddabathuni, col. 4, lines 37 et seq.

Hence, it is clear that the Examiner has cited no portion of the Siddabathuni reference which teaches or suggests “the host of the source identifying to the sending agent [of the source] the data addressed by the virtual memory addresses provided by the sending agent [of the source]” as required by claim 1.

Accordingly, even if it were obvious to combine the Porterfield, Tang and Siddabathuni references, a point not conceded, it is clear that such a combination would fail to meet the required combination of features as recited in claim 1.

It is respectfully submitted that the deficiencies of the Examiner’s citations to the Porterfield, Tang and Siddabathuni references are not met by the Examiner’s citations to the AAPA or Dunham reference. Independent claims 11 and 21 may be distinguished in a similar fashion. With respect to claim 21, a “system adapted to communicate with a destination” is recited in which the system comprises, *inter alia*, a host and a sending agent having the recited internal virtual memory addressing scheme.

The remaining claims depend either directly or indirectly from the independent claims. Accordingly, the rejection of these claims is improper for the reasons given above. Moreover, the dependent claims include additional limitations, which in combination with the base and intervening claims from which they depend provide still further grounds of patentability over the cited art.

It is therefore respectfully submitted that the rejections of the claims should be withdrawn.

Conclusion

Applicants have not added any claims. Nonetheless, should any additional fees be required, please charge Deposit Account No. 50-0585.

The attorney of record invites the Examiner to contact him at (310) 553-7977 if the Examiner believes such contact would advance the prosecution of the case.

Dated: September 11, 2008

By: /William Konrad/

William K. Konrad  
Registration No. 28,868

Please direct all correspondences to:

William K. Konrad  
Konrad Raynes & Victor, LLP  
315 South Beverly Drive, Ste. 210  
Beverly Hills, CA 90212  
Tel: (310) 553-7970  
Fax: 310-556-7984